CTL – Computation Tree Logic
a Logic for Branching-time Model Checking

Paul Jackson
Paul.Jackson@ed.ac.uk
University of Edinburgh

Formal Verification
Autumn 2023

\(^1\)Including contributions by Jacques Fleuriot, Bob Atkey and Elizabeth Polgreen
CTL Syntax

Assume some set $\text{Atom}$ of atomic propositions

$$\phi, \psi ::= p \mid \neg\phi \mid \phi \land \psi \mid \phi \lor \psi \mid \phi \Rightarrow \psi \mid \ax \phi \mid \ex \phi \mid \af \phi \mid \ef \phi \mid \ag \phi \mid \eg \phi \mid \at\phi \mid \et\phi$$

where $p \in \text{Atom}$

Each temporal connective is a pair of a path quantifier

$A$ — for all paths

$E$ — there exists a path

and an LTL-like temporal operator $X, G, F$ or $U$

Precedence high-to-low:
$(AX, EX, AF, EF, AG, EG, \neg),$
$(\land, \lor),$
$\Rightarrow$
CTL Semantics 1: Transition Systems and Paths

(This is the same as for LTL)

Definition (Transition System)

A transition system $\mathcal{M} = \langle S, \rightarrow, L, I \rangle$ consists of

- $S$ set of states
- $\rightarrow \subseteq S \times S$ transition relation
- $L : S \rightarrow \mathcal{P}(\text{Atom})$ labelling function
- $I \subseteq S$ set of initial states (sometimes)

such that $\forall s. \exists t. s \rightarrow t$.

Definition (Path)

A path $\pi$ in a model $\mathcal{M} = \langle S, \rightarrow, L, I \rangle$ is an infinite sequence of states $s_0, s_1, \ldots$ such that $s_0 \in I$ and $\forall i \geq 0. s_i \rightarrow s_{i+1}$. 
CTL Semantics 2: Satisfaction relation

Satisfaction relation $\mathcal{M}$, $s |= \phi$ read as

“state $s$ in model $\mathcal{M}$ satisfies CTL formula $\phi$”.

The $\mathcal{M}$ is often implicit.

- $s |= T$
- $s \not|= \bot$
- $s |= p$ iff $p \in L(s)$
- $s |= \neg \phi$ iff $s \not|= \phi$
- $s |= \phi_1 \land \phi_2$ iff $s |= \phi_1$ and $s |= \phi_2$
- $s |= \phi_1 \lor \phi_2$ iff $s |= \phi_1$ or $s |= \phi_2$
- $s |= \phi_1 \Rightarrow \phi_2$ iff $s |= \phi_1$ implies $s |= \phi_2$
CTL Semantics 3: Satisfaction relation (continued)

\[ s \models \text{AX} \phi \iff \forall s'. s \rightarrow s' \text{ implies } s' \models \phi \]
\[ s \models \text{EX} \phi \iff \exists s'. s \rightarrow s' \text{ and } s' \models \phi \]
\[ s \models \text{AG} \phi \iff \forall \text{ paths } \pi \text{ s.t. } s_0 = s. \forall i. s_i \models \phi \]
\[ s \models \text{EG} \phi \iff \exists \text{ path } \pi \text{ s.t. } s_0 = s. \forall i. s_i \models \phi \]
\[ s \models \text{AF} \phi \iff \forall \text{ paths } \pi \text{ s.t. } s_0 = s. \exists i. s_i \models \phi \]
\[ s \models \text{EF} \phi \iff \exists \text{ path } \pi \text{ s.t. } s_0 = s. \exists i. s_i \models \phi \]
\[ s \models \text{A} [\phi_1 U \phi_2] \iff \forall \text{ paths } \pi \text{ s.t. } s_0 = s.
\quad \exists i. s_i \models \phi_2 \text{ and } \forall j < i. s_j \models \phi_1 \]
\[ s \models \text{E} [\phi_1 U \phi_2] \iff \exists \text{ path } \pi \text{ s.t. } s_0 = s.
\quad \exists i. s_i \models \phi_2 \text{ and } \forall j < i. s_j \models \phi_1 \]
For every next state, $\phi$ holds.
There exists a next state where $\phi$ holds.
For all paths, there exists a future state where $\phi$ holds.
There exists a path with a future state where $\phi$ holds.
For all paths, for all states along them, $\phi$ holds.
There exists a path such that, for all states along it, $\phi$ holds.
For all paths, \( \psi \) eventually holds, and \( \phi \) holds at all states earlier.

\[
A[\phi \mathbf{U} \psi]
\]
CTL in Pictures

\[ E[\phi \ U \ \psi] \]

Exists path where \( \psi \) eventually holds, and \( \phi \) holds at all states earlier.
CTL Examples

- **EF p**
  There exists a path along which p eventually holds

- **AG AF p**
  In all future states, it is always the case that p eventually holds

- **AG (p ⇒ AF q)**
  In all future states, if p holds then always eventually q holds
CTL Examples (continued)

- **AG (p ⇒ E[p U q])**
  In all future states, if p holds then there exists a path onwards along which p continues to hold until q holds

- **AG (p ⇒ EG q)**
  In all future states, if p holds then there exists a path onwards along which p holds forever

- **EF AG p**
  There exists some future state from which p always holds along all paths
CTL Equivalences

demotion dualities for the temporal connectives

\[ \neg \text{EX } \phi \equiv \text{AX } \neg \phi \]
\[ \neg \text{EF } \phi \equiv \text{AG } \neg \phi \]
\[ \neg \text{EG } \phi \equiv \text{AF } \neg \phi \]
\[ \neg \text{EX } \phi \equiv \text{AX } \neg \phi \]

Also have

\[ \text{AF } \phi \equiv A[\top U \phi] \]
\[ \text{EF } \phi \equiv E[\top U \phi] \]
\[ A[\phi_1 U \phi_2] \equiv \neg (E[\neg \phi_2 U (\neg \phi_1 \land \neg \phi_2)] \lor \text{EG } \neg \phi_2) \]

From these one can show that sets \( \{ \text{AU, EU, EX} \} \) and \( \{ \text{EG, EU, EX} \} \) are both adequate sets of temporal connectives.
Differences between LTL and CTL

- LTL allows for questions of form
  - For all paths, does LTL property $\phi$ hold?
  - Does there exist a path on which LTL property $\phi$ holds?
    (Ask whether $\neg \phi$ holds on all paths and look for a counter-example)

- CTL allows mixing of path quantifiers
  - $AG (p \Rightarrow EG q)$

- Some path properties are impossible to express in CTL.
  - In LTL: $GF p \Rightarrow GF q$
  - In CTL: $AG AF p \Rightarrow AG AF q$
    is not the same.
    (Consider a model in which $p$ holds infinitely often on some paths, but not all, and $q$ holds nowhere)
  - Core issue: $\Rightarrow$ in CTL cannot be used to restrict paths

- Exist Fair CTL refinements of CTL that address this issue to some extent
  - E.g. path quantifiers can be restricted to consider only paths on which given properties hold infinitely often.
Fairness

- Key in modelling concurrent systems
- Concurrency handled using Interleaving:

\[(s_1, s_2) \rightarrow (s'_1, s'_2) \equiv (s_1 \xrightarrow{1} s'_1 \land s_2 = s'_2) \lor (s_1 = s'_1 \land s_2 \xrightarrow{2} s'_2)\]

- But want to avoid considering paths in which only one process ever runs
- E.g. in LTL prove properties of form

\[\text{Fair} \Rightarrow \phi\]

where

\[\text{Fair} = (\text{GF taken}_1) \land (\text{GF taken}_2)\]

and taken\(_i\) holds at a state of a path if process \(i\) takes a step from that state to the next state.
Further difference between LTL and CTL

The LTL formula

\[ \text{FG} \, p \]

and the CTL formula

\[ \text{AF AG} \, p \]

are not the same.

Exercise: give a model which satisfies one of the formulas but not the other.
CTL*

- Extends both LTL and CTL
- **State** formulas, evaluated in states:
  
  \[ \phi ::= p \mid \neg \phi \mid \phi \land \phi \mid \phi \lor \phi \mid \phi \Rightarrow \phi \mid A[\alpha] \mid E[\alpha] \]

- **Path** formulas, evaluated along paths:
  
  \[ \alpha ::= \phi \mid \neg \alpha \mid \alpha \land \alpha \mid \alpha \lor \alpha \mid \alpha \Rightarrow \alpha \mid X \alpha \mid F \alpha \mid G \alpha \mid \alpha U \alpha \]

- An LTL formula \( \alpha \) is expressed as \( A[\alpha] \) in CTL*
- Harder to model check
Further Reading
